

# Delay-Optimal Data Forwarding in Vehicular Sensor Networks

Okyoung Choi<sup>†</sup>, Seokhyun Kim<sup>†</sup>, Jaeseong Jeong<sup>†</sup>, Hyang-Won Lee<sup>‡</sup> and Song Chong<sup>†</sup>

<sup>†</sup>Department of Electrical Engineering, KAIST

<sup>‡</sup>Department of Internet and Multimedia Engineering, Konkuk University

**Abstract**—Vehicular Sensor Network (VSN) is emerging as a new solution for monitoring urban environments such as Intelligent Transportation Systems and air pollution. One of the crucial factors that determine the service quality of urban monitoring applications is the delivery delay of sensing data packets in the VSN. In this paper, we study the problem of routing data packets with minimum delay in the VSN, by exploiting i) vehicle traffic statistics, ii) anycast routing and iii) knowledge of future trajectories of vehicles such as buses. We first introduce a novel road network graph model that incorporates the three factors into the routing metric. We then characterize the packet delay on each edge as a function of the vehicle density, speed and the length of the edge. Based on the network model and delay function, we formulate the packet routing problem as a Markov Decision Process (MDP) and develop an optimal routing policy by solving the MDP. Evaluations using real vehicle traces in a city show that our routing policy significantly improves the delay performance compared to existing routing protocols.

## I. INTRODUCTION

Recently, Vehicular Sensor Networks (VSNs) have received a great amount of attention as a new solution for monitoring the physical world [10]. In VSNs, vehicles equipped with sensing devices move around an urban area and sense the urban environment periodically. The vehicles use vehicle to vehicle (V2V) or vehicle to infra (V2I) wireless communications to deliver the sensing data to an urban monitoring center. Hence, unlike the traditional sensing system with fixed sensors that experiences limited coverage, the vehicular sensing system can monitor any area where vehicles can reach. Moreover, the vehicular sensor network can be deployed and maintained with relatively low cost since it does not heavily rely on the network infrastructure for sensing data delivery.

Many of the VSN applications such as Intelligent Transportation System (ITS) require frequent updates of sensing information from all over the urban area, and hence it is important to guarantee *timely delivery* of sensing data from *every area* of interest to the urban monitoring center. Such a *coverage guarantee* is rather challenging in VSNs where the links (and thus the routes to destinations) can come and go depending on the mobility of vehicles. For instance, in such a network with intermittent connectivity, a vehicle sometimes has to carry the data while it moves away from the destination. In fact, Delay-Tolerant Networks (DTNs) similarly experience intermittent routes to destinations, and there has been a large body of work that addresses the problem of routing data packets with minimum delay in DTNs [2], [7], [9], [12], [21]. Due to

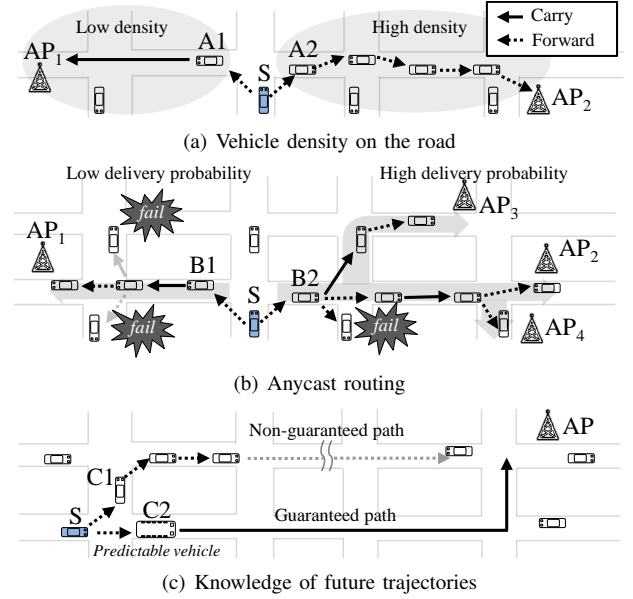


Fig. 1. Important factors on the delay performance in Vehicular Sensor Networks: APs are destinations in anycast routing

the similarity, the packet routing policies for DTNs could be used for VSNs as well. However, the VSN is distinguished from general DTNs in several aspects. First, vehicles in VSNs only move along the road, whereas mobile nodes in general DTNs are typically assumed to be able to move arbitrarily. Second, VSNs generally adopt anycast with multiple destinations, whereas most of the works in general DTNs assume unicast. Third, there are vehicles with predetermined future trajectories, such as buses, whereas in general DTNs, it is hard to predict the movement of mobile nodes. Therefore, the packet routing policies for DTNs may not be directly applicable to VSNs, or may not be able to fully exploit the characteristics of VSNs. In this paper, we study the packet routing problem in the VSN with anycast.

In particular, we focus on minimizing the packet delivery delay from every area of interest to the urban monitoring center. It is obvious that a packet routing algorithm with minimum delay must take into account the aforementioned characteristics of VSNs. First, since the vehicles can move only along the road, the vehicle density can be different from road to road. Clearly, the road with high density can provide more opportunities of

wireless multi-hop transfers, and thus reduce the delivery delay on the road. Consider a source vehicle  $S$  in Fig. 1(a), which tries to select a better relay out of vehicles  $A1$  and  $A2$ . Even though  $A1$  is closer to a destination (or AP in Fig. 1(a)), forwarding to  $A2$  may be more beneficial since the delay of multi-hop transfer over high density road is much smaller than carrying delay. Second, in anycast routing, a data packet just needs to be delivered to any one of the multiple destinations. Hence, the effect of multiple APs can be exploited to reduce the packet delay. As shown in Fig. 1(b), forwarding to  $B1$  can fail to deliver packets to the targeted AP (*i.e.*,  $AP_1$ ) due to the uncertainty in  $B1$ 's movement. However, since there exist many alternative APs on the direction of  $B2$  (*i.e.*,  $AP_2$ ,  $AP_3$  and  $AP_4$ ), forwarding to  $B2$  may be a better option for reducing the delay. Third, the vehicles with known trajectories such as buses can help further reduce the delay. In Fig. 1(c) where  $S$  is far from the destination, such a predictable vehicle  $C2$  guarantees to carry packets to the AP, which can significantly improve the routing performance compared to the delivery along a non-guaranteed path. Note that the effect of known future trajectories is greatly appreciated in the scenario where the vehicle density is relatively low.

Our goal in this paper is to develop a delay-optimal packet routing algorithm in VSNs, by taking into account the above ideas. We first develop a novel road network graph model that incorporates the effect of predetermined vehicle trajectories as well as unpredictable trajectories. This network model is used to characterize the delay on a road segment as a function of the average vehicle density and speed, and the length of the road segment. Based on the network model and delay function, we formulate the routing problem as a Markov Decision Process (MDP) that seeks to minimize the expected delay of a packet from each area to one of the destinations. We develop an optimal packet routing algorithm by solving the MDP. We examine our algorithm using Shanghai vehicle traces [1], [15], and show that the packet delay from each area is significantly improved compared to existing routing algorithms in [8], [16].

The rest of this paper is organized as follows: In Section II, we discuss related work. In Section III, we present the road network graph model of the VSN. In Section IV, we formulate the packet routing problem as an MDP, and develop an optimal routing policy that solves the MDP. In Section V, we evaluate the performance of our routing algorithm using real vehicle traces.

## II. RELATED WORK

There are a number of papers that study packet routing algorithms in VSNs [6], [10], [11] and Vehicular Networks [5], [7]–[9], [17]–[19], [21]. Their common goal is to minimize the packet delivery delay to the destination. The existing routing algorithms can be classified into multi-copy schemes and single-copy schemes.

In multi-copy routing schemes [2], [5], [9], [14], packets are replicated and forwarded to have a better chance of reaching the destination. However, such a replication can result in heavy

congestion, which in turn hinders the packets from reaching the destination. In single-copy routing schemes [7], [12], [19], [21], packets are not replicated, but instead, the characteristics of vehicular networks are better utilized for reducing the packet delivery delay. For instance, the Vehicle-Assisted Data Delivery (VADD) algorithm in [21] makes a routing decision based on the road layout, vehicle density and speed, and is shown to outperform the routing algorithms that do not utilize the characteristics of vehicular networks. The work in [7] improves upon the VADD algorithm by taking into account predetermined trajectories of vehicles. However, they do not fully exploit the knowledge of trajectories in that the entire trajectories and their impact on the delay performance are not incorporated into the routing metric. All of these works assume unicast. In this paper, we investigate the routing problem in VSNs by formulating a Markov Decision Process (MDP) that fully takes into account the effect of predetermined future trajectories and anycast routing as well as the vehicle density and speed.

## III. SYSTEM MODEL

Our vehicular sensor network is modeled as a “vehicular sensing system” working on an urban area or a “road network” described in the following.

### A. Vehicular Sensing System

We consider a Vehicular Sensor Network (VSN) that consists of vehicles and WiFi Access Points (APs). Vehicles moving along the road sense the urban area, generate sensing data packets periodically, and deliver the packets to one of the APs by carrying or forwarding to others. The APs are deployed only at intersections, and connected to the urban monitoring center via wired backhaul networks. Hence, the sensing data packets just need to be delivered to one of the APs. There are two types of vehicles including those with predetermined trajectories (such as buses and police patrol vehicles) and those with unpredictable trajectories (such as taxis and cars). For simplicity of exposition, a vehicle with predetermined trajectory will be called “bus” throughout the paper. As in a real city, we assume that a certain fraction of vehicles in the VSN are buses (*i.e.*, vehicles with predetermined paths).

We assume that vehicles can use the digital roadmap and their GPS information, and are equipped with the IEEE 802.11 devices to communicate with other vehicles or the APs. We also assume that once a vehicle forwards a packet to another vehicle, the packet is immediately deleted from the sender vehicle; so that there is always at most one copy of each data packet in the network.

### B. Road Network Graph

The urban area or road network to be sensed by vehicles is modeled as a graph  $G = (\mathcal{I}, \mathcal{R})$  where  $\mathcal{I}$  is the set of intersections and  $\mathcal{R}$  is the set of road segments connecting the intersections. The network  $G$  is a directed graph, and hence, road segment  $e_{ij} \in \mathcal{R}$  denotes the road from intersection  $i$  to (neighboring) intersection  $j$ . Denote by  $\mathcal{I}_{AP} \subset \mathcal{I}$  the set of

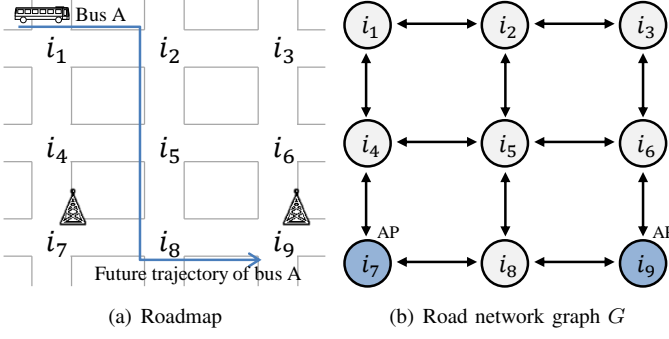


Fig. 2. Roadmap and its corresponding road network graph

intersections where APs are placed. In our system, there are  $N$  vehicles in total, and we define  $\mathcal{V} = \{0, 1, \dots, M\}$  as the set of the types of vehicles. If the type  $v \in \mathcal{V}$  of a vehicle is zero, its trajectories are unpredictable. Otherwise, if  $v \neq 0$ , then it represents a bus line with a predetermined route. Thus,  $M$  is the number of bus lines in the VSN.

Fig. 2 shows the roadmap and its corresponding directed graph  $G$ . In Fig. 2(a), two APs are placed at the intersections  $i_7, i_9$  and the path of bus A is the sequence of intersections,  $i_1, i_2, i_5, i_8$  and  $i_9$ . Note that the path of a packet is a sequence of consecutive road segments and intersections since the packets are carried and forwarded by vehicles moving along the road. Unlike the usual communication network where there is a fixed set of routes all the time, in the VSN, the links on a “data path” are formed by the mobility of vehicles, and thus, they do not always exist. Accordingly, the road network graph  $G$  represents a network that can be “potentially” used for delivering data packets, and the existence of data links in the network is highly uncertain. Therefore, the data packets are delivered as if they are routed over a random graph, and this will be accounted for in our formulation of the packet routing problem in Section IV.

To incorporate the effect of buses into the graph, we note that a bus can carry its packets not only to the neighbor intersections but also to every intersection along its future trajectory with 100% probability in a certain time. Hence, it is as if there is an edge directly connecting an intersection to another intersection which is multiple blocks away. To define these additional edges, we introduce a new notation  $e_{ij}^v$  representing the edge from  $i$  to  $j$  created by type- $v$  vehicle. We denote the set of newly added edges by  $\mathcal{L}$  and a new road network graph by  $G' = (\mathcal{I}, \mathcal{R}')$  where  $\mathcal{R}' = \mathcal{R} \cup \mathcal{L}$ . Note that edge  $e_{ij}^0 \in \mathcal{R}'$  is the same as  $e_{ij} \in \mathcal{R}$  of graph  $G$ . Let  $\mathcal{R}'_s$  be the set of edges in  $\mathcal{R}'$  corresponding to a “single” road segment in  $\mathcal{R}$ , i.e.,  $\mathcal{R}'_s = \{e_{ij}^v \in \mathcal{R}' : \exists e_{ij} \in \mathcal{R}\}$ . Fig. 3 shows the new road network graph  $G'$ , which is augmented from  $G$  in Fig. 2(b) to take into account the effect of bus A’s predetermined path. Using the graph  $G'$ , we formulate the delay-optimal packet routing problem in Section IV.

#### IV. DELAY-OPTIMAL ROUTING ALGORITHM

In this section, we develop a routing policy that minimizes the packet delay to any one of the APs. In particular, we

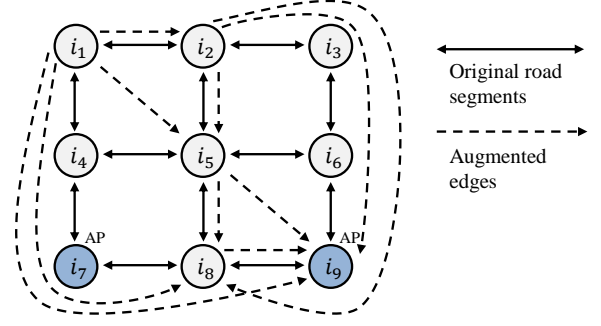


Fig. 3. Augmented road network graph  $G'$  incorporating bus line A

formulate the packet routing problem as a Markov Decision Process (MDP) and find an optimal routing policy that solves the MDP. We also discuss how vehicle traffic statistics can be used to estimate the parameters in the MDP such as state transition probability and (link delay) cost.

##### A. Routing Algorithm Overview

As mentioned in Section III, packets are delivered by vehicles along the intersections and edges in the augmented road network graph  $G'$ . We assume that the routing policy is computed in advance using the vehicle traffic statistics, and the vehicles only have a routing table that can be used for forwarding packets. This would reduce the amount of online computations and thus enable fast forwarding of packets. Our routing algorithm specifies the forwarding decision at every intersection and edge as follows:

1) *At intersections:* Consider a vehicle arriving at an intersection, and assume that it has data packets. Clearly, the vehicle can forward its packets to a neighbor intersection if it meets another vehicle heading to the neighbor intersection or if it moves to the neighbor intersection. As mentioned above, a precomputed routing policy is loaded on the vehicles, and hence, a routing policy that specifies only a single best next hop does not work in our setting. Note that if the edge selected as a single best next hop is not available due to no vehicles moving along the edge, such a routing cannot continue simply because it does not define what to do in that case. Consequently, a contingency decision-making framework is necessary.

Our idea is to prioritize the outgoing edges of each intersection. Thus, if the vehicle does not either meet another vehicle along the edge with the first priority or move onto the edge, then it attempts packet forwarding toward the edge with the second priority, and so on. In Section IV-B, we develop a prioritization method (i.e., routing policy) that minimizes the packet delay.

2) *On edges:* The packet forwarding on an edge, say  $e_{ij}^v$ , is divided into two cases. If  $j$  is a neighbor intersection of  $i$  in the original network graph  $G$  (i.e.,  $e_{ij}^v \in \mathcal{R}'_s$ ), then a vehicle on  $e_{ij}^v$  forwards its packets to a vehicle closer to  $j$ . If  $j$  is not a neighbor intersection of  $i$  in  $G$  (i.e.,  $e_{ij}^v \in \mathcal{R}' \setminus \mathcal{R}'_s$ ), then  $e_{ij}^v$  is an augmented edge by the bus with type  $v$ . On those edges, the bus with the corresponding type carries packets to  $j$ .

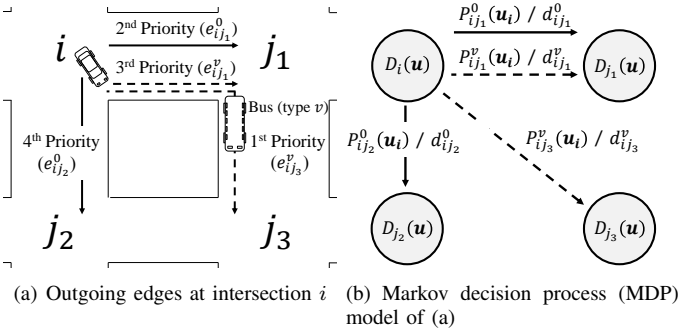


Fig. 4. Formulation of routing problem in graph  $G'$

### B. MDP Formulation and Optimal Routing Policy

Markov Decision Process (MDP) provides a mathematical framework for modeling the decision-making system where the transition from a state to another state is probabilistic, and the transition probability depends on the decision taken at the state. Hence, MDP can effectively capture the essence of the routing problem in VSNs where the delivery of a packet from an intersection to another is subject to uncertainty as there may not always exist a vehicle moving in the desired direction. The set of states in our MDP represents the set of intersections  $\mathcal{I}$ , and the transitions from one state to others occur probabilistically over the edges  $e_{ij}^v \in \mathcal{R}'$ . Then, the control decision at each state in the MDP corresponds to a routing decision at each intersection. Note that the state transition probability from state  $i$  to  $j$  depends on the vehicle traffic statistics and the routing decision at intersection  $i$ . Denote by  $\mathbf{u}_i$  a routing decision at intersection  $i$ . To account for the prioritization discussed above,  $\mathbf{u}_i$  is defined as a row vector,  $\mathbf{u}_i = [u_i^1 \ u_i^2 \ \dots \ u_i^{K_i}]$ , where  $u_i^1, u_i^2, \dots, u_i^{K_i} \in \mathcal{R}'$  are all the outgoing edges from intersection  $i$  and  $K_i$  is the total number of outgoing edges from  $i$ . The order of elements in  $\mathbf{u}_i$  represents the priority, that is,  $u_i^k$  indicates the  $k$ -th most preferred next hop from intersection  $i$  for minimum packet delay. Let  $\mathcal{U}(i)$  be the set of all possible decisions  $\mathbf{u}_i$  at intersection  $i$ , then the size of  $\mathcal{U}(i)$  is given by  $|\mathcal{U}(i)| = K_i!$ .

As mentioned above, the routing decision  $\mathbf{u}_i$  affects the data forwarding probability from intersection  $i$  to other intersections. Let  $P_{ij}^v(\mathbf{u}_i)$  be the probability that a packet is forwarded from intersection  $i$  to  $j$  by a type- $v$  vehicle under a routing decision  $\mathbf{u}_i$ . Denote by  $d_{ij}^v$  the expected data delay on an edge  $e_{ij}^v \in \mathcal{L}$  which is the time it takes to carry and forward a packet along the edge  $e_{ij}^v$ . The delay  $d_{ij}^v$  can be estimated using the average vehicle speed, density and the total length of road segments from intersection  $i$  to  $j$ . We discuss the details of the estimation of  $d_{ij}^v$  in Section IV-D.

Under a routing policy  $\mathbf{u} = [\mathbf{u}_i, \forall i \in \mathcal{I}]$ , let  $D_i(\mathbf{u})$  be the expected data delivery delay from intersection  $i$  to any AP. Thus, for an intersection  $i$  where an AP is placed (i.e.,  $i \in \mathcal{I}_{AP}$ ),  $D_i(\mathbf{u}) = 0$  for any routing policy  $\mathbf{u}$  and no action is taken at  $i$ . On the other hand, at an intersection  $i$  where there is no AP,  $D_i(\mathbf{u})$  depends on  $d_{ij}^v$ ,  $D_j(\mathbf{u})$  and  $P_{ij}^v(\mathbf{u}_i)$  for every

outgoing edge  $e_{ij}^v$  from  $i$ . For better understanding, in Fig. 4, we illustrate an example of the routing decision  $\mathbf{u}_i$  and related parameters. Assume that there exist four possible forwarding candidates which are prioritized by  $\mathbf{u}_i$  as in Fig. 4(a). Based on the routing scenario, the MDP model in Fig. 4(b) has four outgoing edges which correspond to the forwarding candidates, and specifies the forwarding probability  $P_{ij}^v(\mathbf{u}_i)$  and the edge delay  $d_{ij}^v$ . Clearly,  $D_i(\mathbf{u})$  can be computed as

$$D_i(\mathbf{u}) = P_{ij_1}^0 \times (d_{ij_1}^0 + D_{j_1}(\mathbf{u})) + P_{ij_2}^0 \times (d_{ij_2}^0 + D_{j_2}(\mathbf{u})) + P_{ij_1}^v \times (d_{ij_1}^v + D_{j_1}(\mathbf{u})) + P_{ij_2}^v \times (d_{ij_2}^v + D_{j_2}(\mathbf{u})) \quad (1)$$

In general,  $D_i(\mathbf{u})$  can be expressed as follows:

$$D_i(\mathbf{u}) = \sum_{v \in \mathcal{V}} \sum_{j \in \mathcal{I}} P_{ij}^v(\mathbf{u}_i) \cdot (d_{ij}^v + D_j(\mathbf{u})), \quad i \in \mathcal{I} \setminus \mathcal{I}_{AP}. \quad (2)$$

Hence, our routing problem can be formulated as

$$\min_{\mathbf{u}} D_i(\mathbf{u}), \forall i. \quad (3)$$

The optimal solution  $\mathbf{u}^*$  to the above problem gives a routing policy that minimizes the expected delay from  $i$  to any one of the APs. The routing problem can be solved using the value iteration method [4] (see Algorithm 1). For a given initial delay vector  $\mathbf{D}^0$ , the expected delay from intersection  $i$  is updated as in (5). The iteration is terminated if the two consecutive delay vectors  $\mathbf{D}^k$  and  $\mathbf{D}^{k-1}$  are close enough, i.e.,

$$\max_{i \in \mathcal{I}} |D_i^k - D_i^{k-1}| < \epsilon \quad (4)$$

where  $\epsilon$  is a predetermined threshold value. It is known that for each  $i$  the sequence  $\{D_i^k\}$  generated by the iteration in (5) converges close to its optimal value  $D_i^* = D_i(\mathbf{u}^*)$  after a sufficient number of iterations [3]. The optimal routing policy  $\mathbf{u}^* = [\mathbf{u}_i^*, \forall i \in \mathcal{I}]$  is then computed using the estimated optimal delay vector  $\mathbf{D}^k = [D_i^k, \forall i \in \mathcal{I}]$ , as in (6).

#### Algorithm 1 Routing Policy Computation

- 1) **Procedure** ComputingOptimalPolicy( $\mathbf{D}^0$ )
- 2)   **Input:** initial value  $\mathbf{D}^0 = [D_i^0, \forall i \in \mathcal{I}]$
- 3)   **Output:** optimal routing policy  $\mathbf{u}^* = [\mathbf{u}_i^*, \forall i \in \mathcal{I}]$
- 4)   **Local variable:**  $k = 0$
- 5)   **repeat**
- 6)        $D_i^{k+1} = \min_{\mathbf{u}_i \in \mathcal{U}(i)} \sum_{v \in \mathcal{V}} \sum_{j \in \mathcal{I}} P_{ij}^v(\mathbf{u}_i) \cdot (d_{ij}^v + D_j^k)$    (5)
- 7)        $k = k + 1$
- 8)   **until**  $\max_{i \in \mathcal{I}} |D_i^k - D_i^{k-1}| < \epsilon$
- 9)    $\mathbf{u}_i^* = \arg \min_{\mathbf{u}_i \in \mathcal{U}(i)} \sum_{v \in \mathcal{V}} \sum_{j \in \mathcal{I}} P_{ij}^v(\mathbf{u}_i) \cdot (d_{ij}^v + D_j^k), \forall i \in \mathcal{I}$    (6)
- 10) **return**  $\mathbf{u}^*$

**Remark:** In our anycast setting, multiple APs are deployed at intersections, and each intersection  $i$  with an AP will have  $D_i(\mathbf{u}) = 0$ . Thus, the optimal routing policy solving the MDP

would try to forward the packets toward one of the intersections with APs. Therefore, our routing policy can take advantage of multiple destinations in anycast routing.

### C. Data Forwarding Probability $P_{ij}^v(\mathbf{u}_i)$

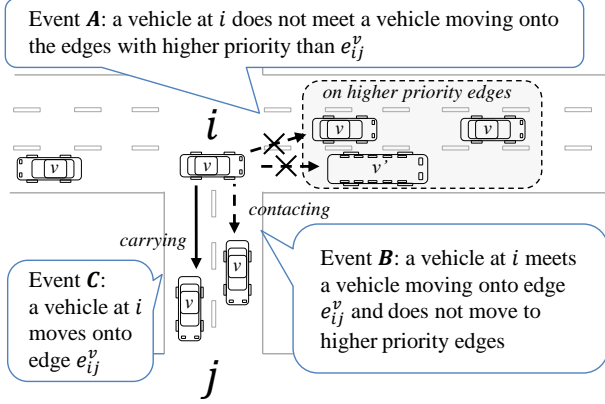


Fig. 5. Conditions for the forwarding of a packet from  $i$  to  $j$ .

Next, we discuss how to calculate the data forwarding probability  $P_{ij}^v(\mathbf{u}_i)$ . Let  $\mathcal{O}(i)$  be the set of outgoing edges from intersection  $i$ . At an intersection  $i$  where there is no AP, a vehicle forwards or carries packets to a neighbor intersection along one of the edges in  $\mathcal{O}(i)$ . Thus,  $P_{ij}^v(\mathbf{u}_i)$  is a function of the probabilities  $Q_{ij}^v$  and  $C_{ij}^v$  defined as

- $Q_{ij}^v$ : probability that a vehicle at  $i$  moves onto edge  $e_{ij}^v$
- $C_{ij}^v$ : probability of contacting a vehicle moving onto  $e_{ij}^v$ .

First, we find an expression for  $P_{ij}^v(\mathbf{u}_i)$  in terms of  $Q_{ij}^v$  and  $C_{ij}^v$ , and then describe how to estimate  $Q_{ij}^v$  and  $C_{ij}^v$  using vehicular traffic statistics. This will provide a complete description of the computation of  $P_{ij}^v(\mathbf{u}_i)$ .

1) *Computation of  $P_{ij}^v(\mathbf{u}_i)$* : Consider an event that packets at intersection  $i$  are forwarded to  $j$  through an edge  $e_{ij}^v$  under a routing decision  $\mathbf{u}_i$ . Clearly, this forwarding event can occur if a vehicle with the packets at  $i$  meets another vehicle moving onto  $e_{ij}^v$  or it moves onto  $e_{ij}^v$ . The additional condition for the forwarding event to occur is that a vehicle at  $i$  does not encounter vehicles moving onto the edges with higher priority than  $e_{ij}^v$  in  $\mathbf{u}_i$  and it does not move onto those edges. Those conditions are illustrated by three events in Fig. 5 that are defined as

- **A**: the event that a vehicle at  $i$  does not meet a vehicle moving onto the edges with higher priority than edge  $e_{ij}^v$
- **B**: the event that a vehicle at  $i$  meets another vehicle moving onto  $e_{ij}^v$  and it does not move onto the edges with higher priority than  $e_{ij}^v$
- **C**: the event that a vehicle moves onto  $e_{ij}^v$ .

Then  $P_{ij}^v(\mathbf{u}_i)$  can be expressed as

$$\begin{aligned} P_{ij}^v(\mathbf{u}_i) &= \Pr[\mathbf{A} \cap (\mathbf{B} \cup \mathbf{C})] \\ &= \Pr(\mathbf{A}) \times \Pr(\mathbf{B} \cup \mathbf{C}) \end{aligned} \quad (7)$$

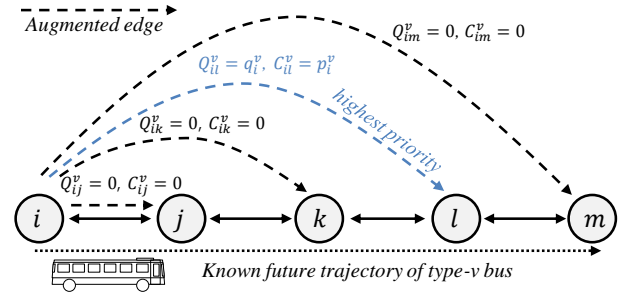


Fig. 6.  $Q_{ij}^v$  and  $C_{ij}^v$  for buses.

where  $\Pr(\mathbf{E})$  is the probability of event  $\mathbf{E}$ . The equality follows from the fact that the moving direction of a vehicle is independent of that of others. Using  $Q_{ij}^v$  and  $C_{ij}^v$ , (7) can be rewritten as follows:

$$\begin{aligned} P_{ij}^v(\mathbf{u}_i) &= \Pr(\mathbf{A}) \times [\Pr(\mathbf{B}) + \Pr(\mathbf{C}) - \Pr(\mathbf{B} \cap \mathbf{C})] \\ &= \Pr(\mathbf{A}) \times [\Pr(\mathbf{B}) + \Pr(\mathbf{C}) - \Pr(\mathbf{B}|\mathbf{C})\Pr(\mathbf{C})] \\ &= \left[ \prod_{e_{ik}^w \in \mathcal{H}(\mathbf{u}_i, e_{ij}^v)} (1 - C_{ik}^w) \right] \\ &\quad \times \left[ C_{ij}^v \left( 1 - \sum_{e_{ik}^w \in \mathcal{H}(\mathbf{u}_i, e_{ij}^v)} Q_{ik}^w \right) + Q_{ij}^v - C_{ij}^v Q_{ij}^v \right] \end{aligned} \quad (8)$$

where  $\mathcal{H}(\mathbf{u}_i, e_{ij}^v)$  is the set of the edges which have higher priority than  $e_{ij}^v$  in a routing decision  $\mathbf{u}_i$ . The first product term in (8) corresponds to  $\Pr(\mathbf{A})$ , i.e., the probability that a vehicle at  $i$  does not meet a vehicle moving onto higher priority edges than the edge  $e_{ij}^v$  in routing decision  $\mathbf{u}_i$ . The second product term is equal to  $\Pr(\mathbf{B} \cup \mathbf{C})$ , i.e., the probability that a vehicle at  $i$  carries or forwards its packets onto edge  $e_{ij}^v$ .

2) *Estimation of  $Q_{ij}^v$  and  $C_{ij}^v$* : Recall that  $Q_{ij}^v$  is the probability that a vehicle at intersection  $i$  moves onto edge  $e_{ij}^v$ , and  $C_{ij}^v$  is the probability of contacting a vehicle moving onto  $e_{ij}^v$ . Obviously,  $Q_{ij}^v$  and  $C_{ij}^v$  are determined by the parameters such as the vehicle density and moving tendency. In particular, the following parameters are used to express  $Q_{ij}^v$  and  $C_{ij}^v$ :

- $q_{ij}^0$ : the fraction of type-0 vehicles moving to a neighbor intersection  $j$  among all vehicles which arrive to  $i$ .
- $p_{ij}^0$ : the probability of meeting a type-0 vehicle at  $i$  that moves to  $j$ .
- $q_i^v$ : the fraction of type- $v$  vehicles among all vehicles which arrive to  $i$ ,
- $p_i^v$ : the probability of meeting a type- $v$  vehicle at  $i$ .

Note that these parameters can be extracted from vehicle traffic statistics [21].

To compute  $Q_{ij}^v$  and  $C_{ij}^v$ , we consider two cases of the vehicle type  $v$ . First, for unpredictable vehicles (i.e.,  $v = 0$ ),  $Q_{ij}^0$  and  $C_{ij}^0$  are estimated as  $q_{ij}^0$  and  $p_{ij}^0$ , respectively. However, in the case of buses (i.e.,  $v > 0$ ), estimating  $Q_{ij}^v$  and  $C_{ij}^v$  is more complicated because at intersection  $i$ , the outgoing edges created by type- $v$  bus become either all available or all

unavailable. Fig. 6 shows an example of such scenarios. If a type- $v$  bus arrives at  $i$ , then all the corresponding outgoing edges become available, in which case only the edge with the highest priority (under a decision  $\mathbf{u}_i$ ) is used. For instance, if the edge from  $i$  to  $l$  has the highest priority among all the edges of type- $v$  bus, the bus carries packets to  $l$ , and hence,  $Q_{ij}^v$  and  $C_{ij}^v$  for other edges (i.e., edges to  $j, k, m$ ) become zero. This shows that for  $v > 0$ ,  $Q_{ij}^v$  and  $C_{ij}^v$  depend on the routing decision  $\mathbf{u}_i$ . Thus, if  $e_{ij}^v$  is the best edge among all augmented edges from type- $v$  vehicles,  $Q_{ij}^v$  and  $C_{ij}^v$  for  $v > 0$  are equal to  $q_i^v$  and  $p_i^v$ , respectively. Otherwise,  $Q_{ij}^v$  and  $C_{ij}^v$  are zero. To describe this as an equation, we introduce a new notation  $\succ_{\mathbf{u}_i}$  such that  $e_{ij}^v \succ_{\mathbf{u}_i} e_{ik}^w$  if  $e_{ij}^v$  has higher priority than  $e_{ik}^w$  under a routing decision  $\mathbf{u}_i$ . The following summarizes the computation of  $Q_{ij}^v$  and  $C_{ij}^v$  for all types of vehicles:

$$Q_{ij}^v(\mathbf{u}_i) = \begin{cases} q_{ij}^v & \text{if } v = 0 \\ q_i^v & \text{if } v > 0 \text{ and } e_{ij}^v \succ_{\mathbf{u}_i} e_{ik}^w, \\ & \forall e_{ik}^w \in \mathcal{O}(i) \text{ s.t. } w = v \\ 0 & \text{otherwise} \end{cases}$$

$$C_{ij}^v(\mathbf{u}_i) = \begin{cases} p_{ij}^v & \text{if } v = 0 \\ p_i^v & \text{if } v > 0 \text{ and } e_{ij}^v \succ_{\mathbf{u}_i} e_{ik}^w, \\ & \forall e_{ik}^w \in \mathcal{O}(i) \text{ s.t. } w = v \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

#### D. Expected Delay on Edges

Recall that  $d_{ij}^v$  is the expected data delay on an edge  $e_{ij}^v \in \mathcal{L}$ . The delay  $d_{ij}^v$  can be estimated using the average vehicle density and speed on  $e_{ij}^v$  and the length of  $e_{ij}^v$ , which can be easily obtained from the vehicle traffic statistics. Note that if  $e_{ij}^v$  corresponds to a set of multiple road segments in  $\mathcal{R}$  (i.e.,  $e_{ij}^v \in \mathcal{R}' \setminus \mathcal{R}'_s$ ), then on  $e_{ij}^v$  the corresponding bus “carries” the packets all the way to intersection  $j$ . On the other hand, if  $e_{ij}^v$  corresponds to a single road segment in  $\mathcal{R}$  (i.e.,  $e_{ij}^v \in \mathcal{R}'_s$ ), then V2V packet forwarding is allowed on  $e_{ij}$  as discussed in Section IV-A. Hence, the delay on the edge is estimated differently depending on the number of hops in the edge.

1)  $d_{ij}^v$  on  $e_{ij}^v \in \mathcal{R}'_s$ : Again, on this type of edges, a data packet is forwarded to another vehicle ahead. Clearly, this V2V forwarding can significantly reduce the delay. The delay  $d_{ij}^v$  depends on the vehicle density  $\rho_{ij}$  on  $e_{ij}^v$  since if the density is high, then there is a high chance of V2V forwarding. Note that if the WiFi transmission range  $R$  is long, then there is also a high chance of V2V forwarding. However, if a vehicle on  $e_{ij}^v$  does not meet another vehicle in the transmission range, then it has to carry its packets all the way to intersection  $j$ . These factors can be integrated in several ways. In this paper, we adopt the delay model in [21] as follows:

$$d_{ij}^v = (1 - e^{-R \cdot \rho_{ij}}) \cdot \frac{l_{ij} \cdot c}{R} + e^{-R \cdot \rho_{ij}} \cdot \frac{l_{ij}}{s_{ij}}, \text{ for } v = 0 \quad (10)$$

where  $l_{ij}$ ,  $s_{ij}$  and  $c$  are the length of road segment  $e_{ij}$ , the average vehicle speed on  $e_{ij}$  and the wireless transmission delay, respectively. The first term in (10) is the expected delay contributed by V2V forwarding, and the second term is the expected carrying delay. The expression in (10) shows that the

delay decreased as the transmission range  $R$ , vehicle density or vehicle speed  $s_{ij}$  increases.

2)  $d_{ij}^v$  on  $e_{ij}^v$  in  $\mathcal{R}' \setminus \mathcal{R}'_s$ : In this case, packets are carried by the bus all the way. Let  $\mathcal{B}(e_{ij}^v)$  be the set of road segments between intersection  $i$  and  $j$  along the route of type- $v$  bus. The packet delay on  $e_{ij}^v$  depends only on the average speed of the bus with type  $v$ , denoted by  $s_{ij}^v$ , and the length of road segments in  $\mathcal{B}(e_{ij}^v)$ . Hence, the expected delay on edge  $e_{ij}^v$  can be estimated by the following equation:

$$d_{ij}^v = \sum_{e_{ij} \in \mathcal{B}(e_{ij}^v)} \frac{l_{ij}}{s_{ij}^v}, \text{ for } v > 0 \quad (11)$$

**Remark:** The delay function clearly shows that our routing algorithm would prefer to forward packets along the edges with high density and high average vehicle speed.

## V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our routing algorithms based on real traces: Shanghai taxi [15] and bus [1] trace. The results show that our routing algorithms improve the delay performance by up to 105% against the existing algorithm [8] in terms of the ratio of packets delivered to destinations in reasonable time.

### A. Simulation Setup

To verify our optimal routing algorithms, we use GPS traces of 4800 taxis [15] and 2300 buses [1] in Shanghai, where the location information of each vehicle is recorded at every 30 seconds in 30km×30km Shanghai for 28 days. To focus on the sensing scenario of downtown area, we select 4.5km×4km Shanghai downtown, which consists of 84 intersections and 112 road segments, as shown in Fig. 7. The selected area is modeled as a road network graph as discussed in Section III. Fig. 7 also shows 5 candidate intersections where APs can be placed (see the intersections with dotted circle). The algorithms will be evaluated for various numbers of APs. We choose 120 taxis and 80 buses (6 types of buses) which have relatively low GPS errors and move within the area of our interest for more than 20 minutes during one hour period from 10AM to 11AM. Since vehicles in Shanghai traces usually move outside the selected area, we define “effective number of vehicles” as the average number of vehicles which stay in the selected area during simulation, and use it instead of the number of all vehicles when describing our simulation setup. We denote by  $N'$  the effective number of vehicles. Note that  $N'$  can be viewed as the vehicle density.

We implement the vehicular sensor network on a well-known wireless network simulator, GloMoSim [20], using 802.11a MAC layer protocol<sup>1</sup>. In the VSN, every vehicle moves along the road and senses the urban area. The vehicles are assumed to generate sensing data packets when they satisfy at least one of the following conditions: 1) a vehicle moves 100m

<sup>1</sup>Although the last update year of GloMoSim was 2001, the IEEE 802.11a MAC protocol stack of GloMoSim is still consistent to the current standards and many of physical layer models are included [13].



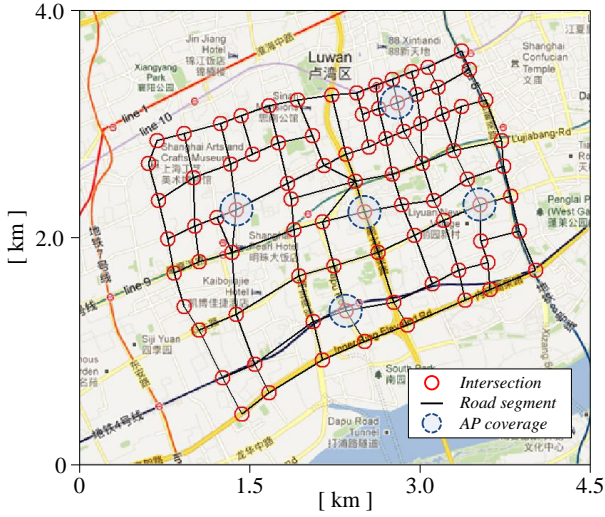


Fig. 7. Road network topology and AP deployment in Shanghai urban area

without any data generation. 2) a vehicle moves without any data generation during 30 seconds. 3) a vehicle reaches at one of 84 intersections. Thus, vehicles generate packets based on their moving distance, time and geographic position. All the vehicles and APs periodically send beacon packets to detect each other every second. If they detect each other, they try to send their packets based on their routing algorithms. Table I presents parameters and scenarios.

TABLE I  
SIMULATION SETTINGS

Parameter Name	Definition
Effective number of vehicles (buses)	95(30), 80(27), 65(22), 55(18), 45(14)
Number of bus lines	6
Number of APs	1, 3, 5
Simulation time	1 hour
Wireless device	802.11a
Data packet size	512 Bytes
Communication range	150m

### B. Tested Algorithms

We test two VSN routing algorithms in the literature: 1) Epidemic routing protocol [16] which floods packets over the network and 2) GPSR [8] which forwards packets to a node located closer to a destination. In anycast version of GPSR, the forwarding decision at an intersection is made by comparing each neighbor intersection's distance from its closest AP. Further, our version of GPSR adopts buffer in each vehicle, so that when vehicles fail to transmit packets, they store the packets in the buffer and keep moving around. We consider two versions of our optimal routing algorithm: (i) Optimal VSN Data Forwarding that regards all the vehicles as those with unpredictable trajectories (denoted by OVDF-U), and (ii) Optimal VSN Data Forwarding that takes into account, the vehicles with known future trajectories (denoted by

OVDF-P). The forwarding decision in OVDF-U is computed assuming that the vehicle type set  $\mathcal{V} = \{0\}$ , even if there are buses moving in the network. On the other hand, the algorithm OVDF-P fully exploits the known trajectories by incorporating them into the formulation through the augmented network graph. We will study the effect of utilizing the vehicles with known trajectories. Note that other routing algorithms for unicast cannot be directly applied to the anycast setting.

### C. Simulation Results

1) *Sensing Coverage*: We first evaluate the sensing coverage of the four algorithms mentioned above. One of the most important performance metrics in VSN routings is the delivery ratio within a certain deadline. In our simulation, the deadline of a packet is fixed to 10 minutes since its creation. To show the spatial coverage performance, we divide  $4.5\text{km} \times 4.5\text{km}$  of Shanghai downtown into a grid of  $72 \times 0.5\text{km} \times 0.5\text{km}$  squares and measure the delivery ratios of the algorithms for each square. Those are 80 vehicles (including 27 buses) and 3 APs.

Fig. 8 plots the delivery ratio within 10 minutes where the  $x$ - $y$  plane represents the grid and  $z$ -axis represents the delivery of sensing data packets generated in the corresponding squares. Out of all the total packets generated, more than 90% of them are from 28 squares (this is due to the road structure of Shanghai downtown imbalance among squares). Note that the areas that generated too few packets do not give meaningful results. Thus, we only consider those squares in the results and call them the "valid squares."

Compared to GPSR and Epidemic routing, OVDF-U and OVDF-P show higher packet delivery ratios in most of the regions. Especially, as shown in Fig. 8(a) and Fig. 8(c), 11 squares (out of 28 valid squares) in OVDF-P achieve at least 20% higher delivery ratio (up to 105% higher) than those in GPSR, and on average, the delivery ratio gain over GPSR is 22% for all of the valid squares. Compared to OVDF-U, OVDF-P shows 9% higher average delivery ratio, and thus, OVDF-P guarantees the best sensing coverage in this scenario. In Fig. 8(d), Epidemic routing shows the lowest packet delivery ratios in most of the squares, due to heavy congestion caused by packet replications.

An important observation in Fig. 8 is that, for any routing algorithm, most of the packets generated in the regions close to APs are delivered to an AP within 10 minutes. This is because those data are close enough to be forwarded to the AP in one or a small number of hops, and thus, the impact of optimal routing decisions on the delivery ratio is relatively small. On the other hand, the intelligence of routing algorithms have a huge impact on the delivery ratio for data packets created in the edge areas that are far from APs. Note that there are abundant routes from an edge area to the APs, however only a few of them are those with low delay. Consequently, in the edge areas, the delay performance can vary substantially depending on the routing algorithms. Hence to verify the performance of the routing algorithms, it is necessary to evaluate the delivery ratio from distant areas. In following, we study the routing

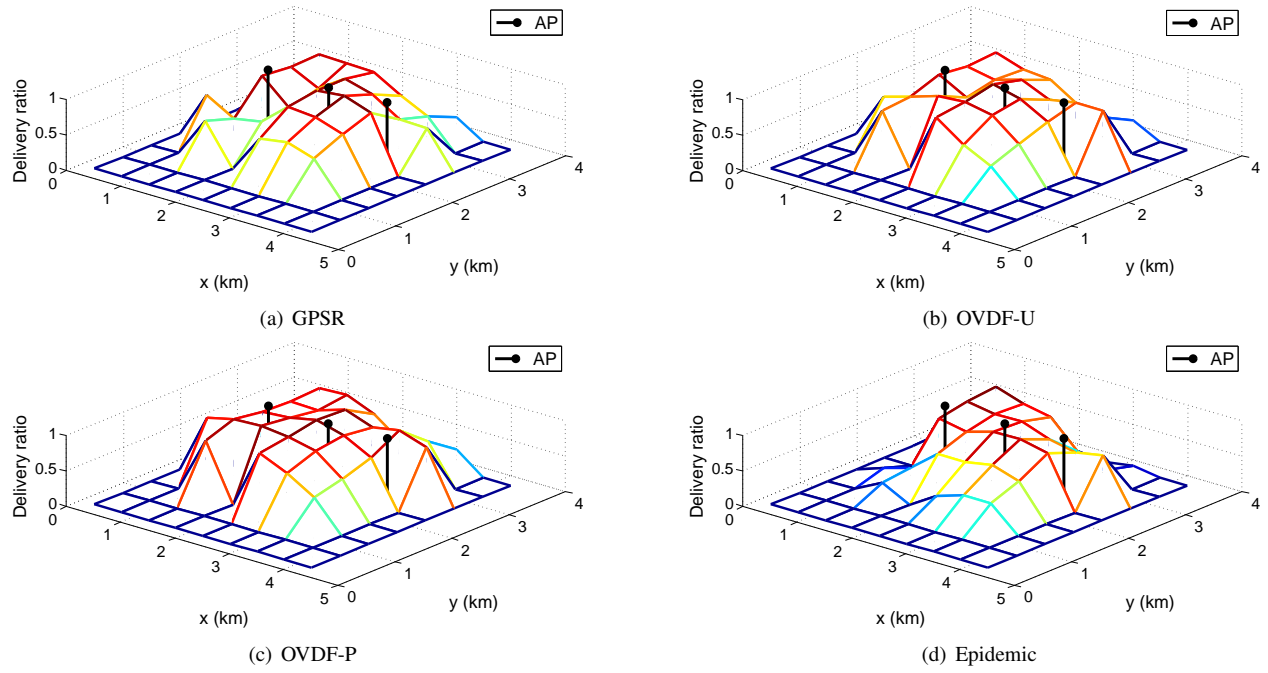


Fig. 8. Vehicular sensing coverage of four tested algorithms : (a) GPSR, (b) OVDF-U, (c) OVDF-P, (d) Epidemic. In all cases, the number of AP is 3 and the number of effective vehicles is 80 (including taxis and 6 different types of buses).

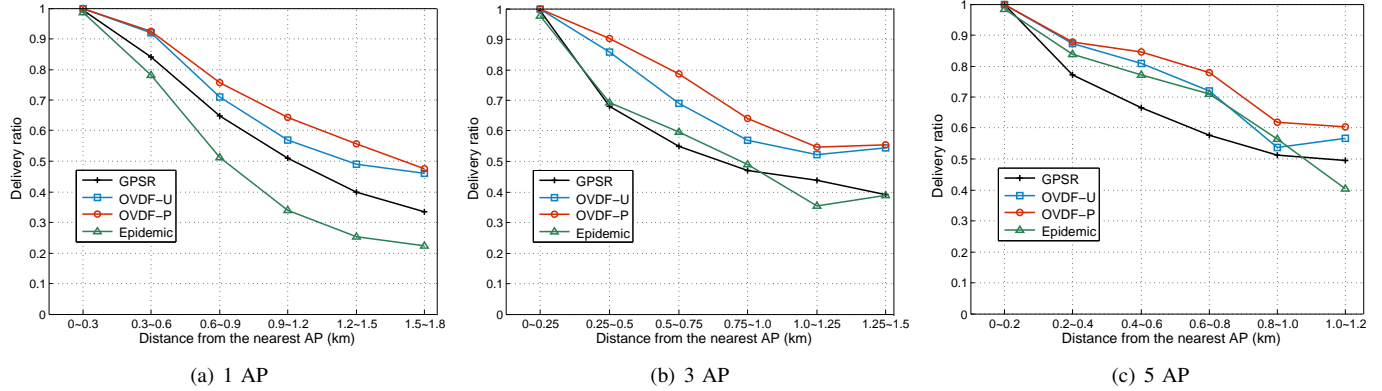


Fig. 9. Delivery ratio with 10 minutes deadline versus distance to AP for the various number of APs: (a) 1 AP, (b) 3 AP, (c) 5 AP. In all cases, the number of effective vehicles is 55 (including taxis and 6 different types of buses).

performance against the distance to APs.

2) *Performance against Distance to APs*: The packet delivery ratio against the distance is estimated as follows: Consider a packet generated at a position  $X$  in the plane, and let  $Y$  be the position of the nearest AP from  $X$ . If the distance between  $X$  and  $Y$  is the range of  $a \sim b$  km, then the delivery of the packet contributes to the delivery ratio from the areas whose nearest APs are  $a \sim b$  km away. There are 55 effective vehicles including 18 buses.

Fig 9 shows that for the regions within the wireless communication range of APs, the delivery ratio is 100%. As the distance between grids and APs increase, the delivery ratio decreases. Note that OVDF-P outperforms GPSR by up to 40%, 43% and 37% for the scenario of 1 AP, 3AP and 5 APs, respectively. For the entire range of the distance, the

performance gap between GPSR and OVDF-U is larger in anycast scenarios (*i.e.*, 3 or 5 APs) than in the unicast scenario (*i.e.*, 1 AP). This is because GPSR is not designed for anycast routing and thus cannot fully exploit the advantage of multiple destinations. Moreover, OVDF-U shows higher delivery ratio than GPSR even in the unicast scenario since we estimate the expected delay on a road segment based on not only its length but also the vehicle density and speed on the road. Clearly, OVDF-P shows the best delivery ratio performance among all the algorithms, especially for the packets generated far from APs.

As shown in Fig. 9(a), in the unicast scenario, Epidemic routing shows poor delivery performance 60% lower than that of OVDF-P at the distance of 1.2km. In anycast scenarios, spreading the packets all over the network can significantly



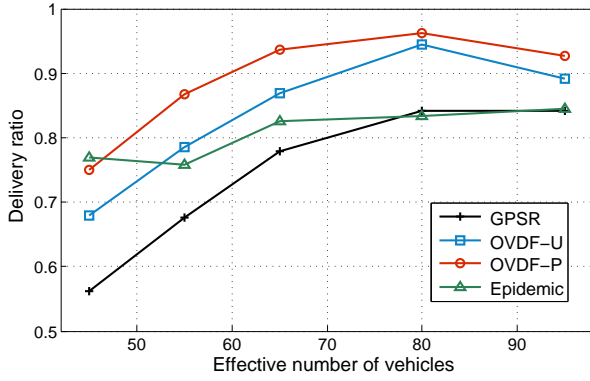


Fig. 10. Data delivery ratio within 10 minutes versus the effective number of vehicles when the number of AP is 5.

improve the delay performance since there are multiple APs throughout the network. On the other hand, such an effect becomes marginal in the unicast scenario where there is only one AP. Moreover, it incurs severe packet collisions as many heavily-loaded vehicles would attempt to transmit to the single AP. This is why Epidemic routing shows poor performance in the unicast scenario.

3) *Performance against the Number of Vehicles*: Next, we compare the performance by changing the number of effective vehicles. The number of APs is 5, and only the packets generated at least 400m away from all the APs are considered. Fig. 10 shows the delivery ratio of each algorithm. The observation from these results are threefold. First, when the density of vehicles in the road network is high, the algorithms achieve comparable performance because, in the densely connected vehicular network, packets are delivered to the destination mostly by V2V forwarding. Second, the performance gap between OVDF-P and other algorithms increases as the density of vehicles decreases. Fig. 10 shows that OVDF-P achieves 11% higher delivery ratio than OVDF-U, when the effective number of vehicles is 45. In sparse vehicular networks, the packet delivery depends heavily on carrying, and thus the effect of vehicles with known trajectories is much more appreciated. Third, in the case of low vehicle density, the delivery ratio of Epidemic routing is relatively high as opposed to the high density scenario because when there are not many routes to destinations due to a small number of vehicles, replicating the packets would greatly improve the chance of reaching a destination.

## VI. CONCLUDING REMARKS

Many of emerging VSN applications require timely delivery of sensing data and wide sensing coverage. However, this is a challenging problem in the VSN where the data links are intermittently connected. To address the issues, we develop a delay-optimal VSN routing algorithm, capturing three key features in urban VSNs: (i) vehicle traffic statistics, (ii) anycast routing and (iii) known future trajectories of vehicles such as bus. Using real traces of 120 taxis and 80 buses in Shanghai we conduct extensive simulations on GloMoSim simulator, and

show that our optimal algorithm outperforms other existing algorithms. Our results demonstrate that carefully designed packet routing algorithms can greatly improve the delay performance in the VSN, and thus are the key to the success of VSN applications that require stringent delay performance guarantee. In this paper, we focused single-copy routing algorithms. Although multi-copy routings can incur serious congestion, there are scenarios where multi-copy routings outperform single-copy routing. Therefore, it would be interesting to extend our framework to account for multi-copy routing, and we leave this as future study.

## REFERENCES

- [1] Shanghai grid. online <http://www.cse.ust.hk/dcrq>.
- [2] A. Balasubramanian, B. N. Levine, and A. Venkataramani. DTN routing as a resource allocation problem. In *Proceedings of SIGCOMM*, 2007.
- [3] R. Bellman. A markovian decision process. *Indiana Univ. Math. J.*, 6:679–684, 1957.
- [4] D. P. Bertsekas. Dynamic programming and optimal control. volume 2. *Athena Scientific 3rd edition*, 2007.
- [5] J. Burgess, B. Gallagher, D. Jensen, and B. Levine. Maxprop: Routing for vehicle-based disruption-tolerant networking. In *Proceedings of IEEE INFOCOM*, 2006.
- [6] Q. Fu, L. Zhang, W. Feng, and Y. Zheng. Dawn: A density adaptive routing algorithm for vehicular delay tolerant sensor networks. In *Communication, Control, and Computing (Allerton)*, 2011 49th Annual Allerton Conference on, pages 1250–1257, sept. 2011.
- [7] J. Jeong, S. Guo, Y. Gu, T. He, and D. Du. Tbd: Trajectory-based data forwarding for light-traffic vehicular networks. In *Distributed Computing Systems, 2009. ICDCS '09. 29th IEEE International Conference on*, pages 231–238, june 2009.
- [8] B. Karp and H. T. Kung. Gpsr: greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking, MobiCom '00*, pages 243–254, New York, NY, USA, 2000. ACM.
- [9] K. Lee, Y. Yi, J. Jeong, H. Won, I. Rhee, and S. Chong. Max-contribution: On optimal resource allocation in delay tolerant networks. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9, march 2010.
- [10] U. Lee, E. Magistretti, M. Gerla, P. Bellavista, and A. Corradi. Dissemination and harvesting of urban data using vehicular sensing platforms. *Vehicular Technology, IEEE Transactions on*, 58(2):882–901, feb. 2009.
- [11] X. Li, W. Shu, M. Li, H. Huang, and M.-Y. Wu. Dtn routing in vehicular sensor networks. In *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, pages 1–5, 30 2008-dec. 4 2008.
- [12] C. Liu and J. Wu. Practical routing in a cyclic mobispace. *IEEE/ACM Trans. Netw.*, 19(2):369–382, Apr. 2011.
- [13] E. Magistretti, O. Gurewitz, and E. W. Knightly. 802.11ec: Collision avoidance without control messages. In *Proceedings of ACM MOBICOM*, 2012.
- [14] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking, WDTN '05*, pages 252–259, New York, NY, USA, 2005. ACM.
- [15] S. J. U. Traffic Information Grid Team, Grid Computing Center. Shanghai taxi trace data. <http://wirelesslab.sjtu.edu.cn/>.
- [16] A. Vahdat and D. Becker. Epidemic routing for partially-connected ad hoc networks. Technical Report, CS-200006, Duke University, April 2000.
- [17] H. Wu, R. Fujimoto, R. Guensler, and M. Hunter. Mddv: a mobility-centric data dissemination algorithm for vehicular networks. In *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks, VANET '04*, pages 47–56, New York, NY, USA, 2004. ACM.
- [18] Y. Wu, Y. Zhu, and B. Li. Trajectory improves data delivery in vehicular networks. In *INFOCOM, 2011 Proceedings IEEE*, pages 2183–2191, april 2011.
- [19] F. Xu, S. Guo, J. Jeong, Y. Gu, Q. Cao, M. Liu, and T. He. Utilizing shared vehicle trajectories for data forwarding in vehicular networks. In *INFOCOM, 2011 Proceedings IEEE*, pages 441–445, april 2011.

- [20] X. Zeng, R. Bagrodia, and M. Gerla. Glomosim: a library for parallel simulation of large-scale wireless networks. In *Parallel and Distributed Simulation, 1998. PADS 98. Proceedings. Twelfth Workshop on*, pages 154 –161, may 1998.
- [21] J. Zhao and G. Cao. Vadd: Vehicle-assisted data delivery in vehicular ad hoc networks. *Vehicular Technology, IEEE Transactions on*, 57(3):1910 –1922, may 2008.